

# Report of WEEK 3

Y Tagore Ravindra Sree

## 1 Searching Algorithms

### 1.1 Linear Search

- Checks each element one by one.
- Works on unsorted lists.
- Time Complexity:  $O(n)$  (worst and average case).

### 1.2 Binary Search

- Works on sorted arrays/lists.
- Uses a divide and conquer approach.
- Time Complexity:  $O(\log n)$  (worst and average case)

### 1.3 Jump Search

- Works on sorted arrays.
- Jumps in fixed steps, then performs linear search. Step size is square root of the target.
- Time Complexity:  $O(\sqrt{n})$

## 1.4 Exponential Search

- Works on sorted arrays.
- Doubles the search range, then performs binary search.
- Time Complexity:  $O(\log n)$

## 1.5 Ternary Search

- Similar to binary search but divides the array into three parts.
- Works on sorted arrays.
- Time Complexity:  $O(\log n)$

## 1.6 Interpolation Search

- Works on sorted, uniformly distributed data.
- Estimates the position of the key (better than binary search for some cases).
- Time Complexity:  $O(\log \log n)$

## 1.7 DFS

- Used for tree/graph traversal.
- Explores as far as possible before backtracking.
- Time Complexity:  $O(V+E)$  (vertices+edges).

## 1.8 BFS

- Also for tree/graph traversal.
- Explores all neighbors at the current level before moving deeper.
- Time Complexity:  $O(V+E)$